A Model to Specify Inter-company Cooperation

Meriem Kermani[1], Mahmoud Boufaida[1] and Emmanuel Paviot-Adet[2]
[1]LIRE laboratory, Constantine 2 University, Constantine, Algeria
[2]LIP6 laboratoty, Pierre & Marie Curie University, Paris, France

Keywords: Interoperability, Inter-company Cooperation, Learning Agents.

Abstract: The impact of the evolution of the global economic ecosystem, has forced and still force again organizations to either adopt new behaviour schemas and to deeply change their structure, also to be more open to their environment as well. These impacts require a new cooperation philosophy from organizations side. Several studies have focused on the problem of the intercompany cooperation, proposing approaches that provide interoperability mechanisms. But this remains an open research domain. In this work we will propose a mediation architecture between different companies. A mediator allows us to create an intercompany cooperative process, the purpose of this solution is to keep the company architecture and ask the mediator which is a software-based agent to play an intermediary role between companies, and involve it in making the transformation between companies. We define a dynamic and cooperative inter-companies model that combines the agent technology and the decision trees paradigm. This last facilitates making decision by selecting the services that best meet customer needs, in order to create a composite service.

1 INTRODUCTION

The company environment has strongly evolved in recent years; it has become more complex and unpredictable, moreover technological changes have greatly influenced the organization of the companies. The integration of these changes becomes essential to ensure companies' survival, not only at a single company, a sector or a given economy level but at global level (Grefen et al., 2009) companies are now trying to "squaring the circle" because they must firstly meet the competitive challenges by improving industrial performance in terms of cost, time, etc. Secondly answer the openness problems. Many companies have realized the importance of cooperation to ensure their survival. Cooperative spaces, in which companies work and react together have emerged in various forms: virtual company, companies' network etc. The establishment and the management of this type of organization are often based on technical and information sharing platforms which support and facilitate the cooperation (Grefen et al., 2009). We will focus in our work in this kind of relationship and more particularly to the notion of the on demand cooperation. We take the definition proposed by (Boukadi et al., 2010) of the on demand cooperation as a temporary grouping of partners distributed in space and time. This grouping is formed from opportunistic alliances initiated by a company called the initiator company of cooperation project which will be dissolved once the opportunity finished. Such a scenario involves the collaboration of different parts of a cooperative process that consists of several processes performed by different partners in order to meet a common purpose or to seize an opportunity in the market.

While the information technology innovates increasingly, but the industrial world is still reluctant to join dynamic cooperation scenarios, since the companies' information system is not suited to this type of operation.

The service-oriented architecture and Web service technology seem to provide credible answers to the needs both internally (information system support of the cooperation) and externally (services available to partners). But in the current state, several obstacles constrain the implementation of the SOA within the company as well as the forming of services-based cooperation. It is basically the lack of implementation method to define the architecture of services within the company information system. In addition, there are few works about the construction methods of the on-demand cooperative process based on services composition paradigm. To these

Kermani M., Boufaida M. and Paviot-Adet E..
A Model to Specify Inter-company Cooperation.
DOI: 10.5220/0004841103180323
In *Proceedings of the 10th International Conference on Web Information Systems and Technologies* (WEBIST-2014), pages 318-323
ISBN: 978-989-758-023-9

two problems a third one related to the dynamic nature of the cooperation is added. The analysis of the problem of services based inter-company cooperation has led us to the decomposition into three additional sub issues that shape our research problem, which are respectively the building problem of a new framework that ensures the effectiveness and efficiency of the intercompany cooperation based on service approach, the problem of decision making in order to choose the best services for a given query and the building problem of the on-demand collaborative process (or the composition of different services). Our contributions answer the limitations and problems previously described by offering mediation architecture between different companies which are a cooperation partners, the use of mediator ensures interoperability between companies and some level of security. To facilitate decision making and be able to choose the services that best meet customer needs, we propose the integration of decision trees in the software agents of our mediator. The third part of our contribution is the implementation of cooperation; this cooperation necessarily passes through an interconnection of processes of different companies, using the service composition paradigm.

This paper is structured as following: Section 2 and 3 are consecrated to a study of our proposal which will be illustrated by an example in section 4. Section 5 is dedicated to a comparative study between multiple research works finally in section 6, we will make conclusions about our work and we will talk about the extension of it.

2 THE PROPOSED MODEL

After having introduced the concept of Inter-enterprise cooperation that results in the formation of business process built from the interconnection of various processes and the partners have a partial master on the business process. We propose in this part mediation architecture between the services of the companies that are participating to the cooperation. According to J-L Lascoux (Touzi et al, 2008) the mediator "provides a framework with its own benchmarks, consisting of operating and communication rules, and steps process. It begins with the acknowledgement of the parts position in terms of legitimacy, to the formalization of an agreement as satisfactory as possible for the parts". In our architecture the mediator is used to perform a dynamic intercompany process. To ensure the interoperability of the participating companies we propose the integration of software agents in our mediator, learning of these agents is done by decision trees. The intercompany process (composite service created) will be published to be used as a web service, to permit the visibility and to be reused by other customers. Figure 1 shows an overview of our architecture.

Figure 1 shows our mediation architecture that consists of:

- Participating Companies: These companies offer accessible services via the public part of their information system. The Information about each service is stored in a services directory of the mediator.
- Mediator: is the core of the architecture, it is acting as an intermediate between the requester and the partners. It is composed of:

- Interface Agent: it receives and makes the processing of the applicant request to build a decision tree, which will be sent to the coordinator agent.
- Services Directory: at this directory we have services agents each of these agent is related to a service proposed by a company. These agents

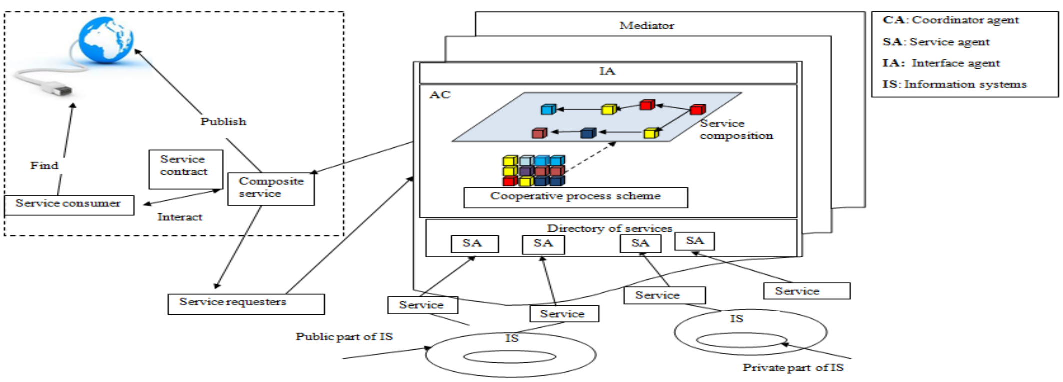

Figure 1: Architecture for intercompany cooperation.

represent information about each service, the information is organized as: a local decision tree in each agent service, and service address, communication protocol, and the details of the service purpose.

- Coordination Agent: it allows creating a composite service to meet the client request. It has a global decision tree that is created from the set of local decision trees. This tree is used to make a decision which allows choosing the best services to meet customer demand.

➢ Applicant service: it represents the launch process.

3 LEARNING AGENTS BY DECISION TREES

To enable the communication of different heterogeneous companies we propose to combine the software agents and decision trees.

Figure 2 shows the general structure of the coordination agent, respecting the three properties of independence, communication and intelligence. This architecture is inspired from the modular theory of (Fodor, 1983) that consists of five modules managing knowledge, perception, communication, control and the thinking of the agent.

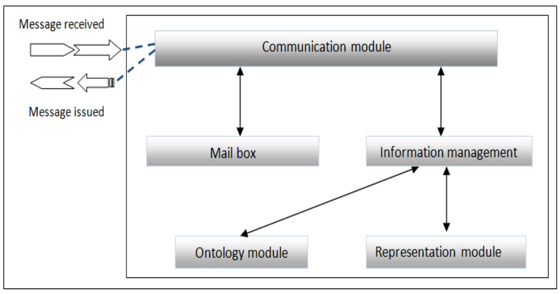

Figure 2: Components of the Coordinator Agent.

➢ Communication Module is a messages support between the agent and its coordinator. All interactions between the coordination agent and the services agent or the interface agent pass through this module.

➢ Mailbox is used to submit messages. This queue box type FIFO (First in First Out), is used for storing messages in order to process them asynchronously.

➢ Representation module receives the request sent by the agent service as a tree, and then reformulates the global tree (the tree that includes all local trees published by the different service agents) following the order of criteria send by interface agent

➢ Information Management Module contains information related to the service published by partners (protocol, address, message structure, description of the roles of services). Each agent service uses a local ontology to represent vocabulary and concepts in its application domain.

➢ Ontology module deals with the search for correspondences between agents to perform a cooperative task.

4 DECISION-MAKING MECHANISM BASED ON DECISION TREES

The coordinator agent uses a decision tree to quickly select the services that best meet customer needs, a decision tree can be used to clarify and find an answer to a complex problem. The structure allows our coordinator agent to take a problem with multiple possible solutions and display it in a simple and easily understandable format that shows the relationship between different events or decisions.

4.1 Structure of the Local Tree

The following figure shows the internal structure of our local trees:

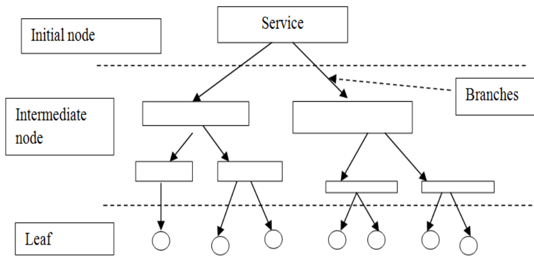

Figure 3: Internal structure of our local trees.

- *Initial Node:* each local tree initial node located in an agent service represents a company service.

-*Branches:* they connect an interior node" parent" to a "child" node. They represent the variable value tested in the parent intermediate node to the child node. Each branch corresponding to a question, allows choosing the path to follow.

Rule 1: the path to go in the tree: if the answer is yes, then the left path will be taken, otherwise it must move to the right one.

-*Intermediate Node:* represents the deals offered by each service, each intermediate node corresponds to a response to determine the deal. Each answer then determines the next question.

Rule 2: each non-terminal node corresponds to a test $P(x)$ which evaluates the predicate P in Example x

-Leaf: each leaf refers to a class.

4.2 Training Algorithms by Decision Tree

The main training algorithms by decision trees are CART (Breiman et al, 1994) and C4.5 (Quinlan, 1984). We present below some of their characteristics. The first step is to build a small tree which consists of the most of data. The idea is to divide recursively and efficiently as possible the training sets examples by defined tests using attributes until we get subsets of examples which contain only examples belonging to the same class. This idea leads to a Top-Down construction methods that's mean constructing the tree from the root to the recursive leaves. We can then define a general scheme of algorithm.

Generic Training Algorithm:

```
BEGIN
| Initialize the current tree to the empty tree; the
| root is the current node
| REPEAT
| | Decide whether the current node is terminal
| | If the node is terminal then
| | | Assign a class to it
| | | Else
| | |Select a test and create as many new son nodes
| | |as there
| | |are possible answers to the test
| | | End if
| | Pass to the next unexplored node if it exists until
| | a decision tree is obtained
| |End
|END
```

5 EXAMPLE: EMERGENCY CASE

We will treat in this part the realization of a response process of an emergency case which is an accident, using three existing entities named: Hospital, SAMU and Police.

- The Hospital service: each hospital propose a description including its position, in our case we will need a GPS coordinates of the hospital.
- The SAMU service: provides a description of its services such as the emergency care team, the possible route and the expected duration of each route.

The Police service: provides the position of the police stations and available patrols.

- The description of each hospital the SAMU and The Police services is published in an agent service and represented by a local decision tree, and the coordination agents representing the global tree for each existing entities.

5.1 Operational Scenario

Using the three services mentioned previously, we will elaborate an emergency response scenario to carry out a sequence of actions for each accident report. We wish to create here a response emergency service which is called 333. This service will allow users to report an accident at a given position. The service will take the necessary steps and return to the user the expected intervention delay in seconds. The internal scenario which will lead the interactions between services is the following:

The coordinator agent will firstly ask the accident base to determine whether the reported accident is known, if so a negative value is simply returned to the service requester. Otherwise, we will first find the nearest hospital to the accident position, we use GPS data of the hospital and we determine both the SAMU and the police to be sent. We then add this to the accidents base before returning to the user the intervention time that was given by the SAMU.

5.2 Creation of Trees

We have chosen to use the SIPINA method for the generalization of our local trees, SIPINA is software but it is also a training method. It generalizes the trees by introducing an additional operation which is the fusion, during the induction of the prediction model. The figure 4 shows the generalization of the tree from Excel files, after following all method steps we obtain the following tree. After creating all local trees, the coordinator agent generates a global tree that includes all local trees; this tree that has been created manually, until the definition of an algorithm which allows the fusion of the local trees according to a test on each arc in order to create a tree that includes other trees by category.

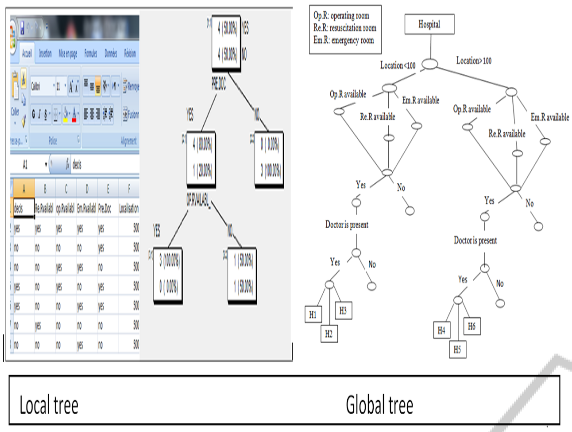

Figure 4: Generalization of the tree from Excel using Sipina Version 3.2.

5.3 Some Implementation Aspects

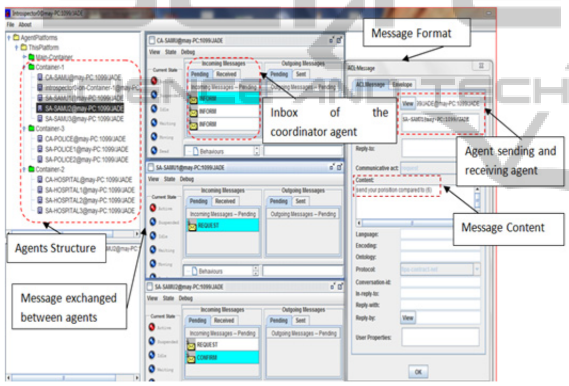

Figure 5: Communication between agents JADE.

Figure 5. shows the structure of our agents, and some messages exchanged between its, using the Jade platform to create the behavior of agents and to examine the feasibility of our proposition. For example coordinator agent of SAMU service named "CA-SAMU" sends a request to the service agents "SA-SAMU" the representatives of the different SAMU, then this services respond with a messages "inform" given their position, the coordinator agent selected the service that best meets the needs and sends a message "confirm" to the service agent concerned, the traces of the messages exchanged between agents are made with the help of sniffer agent. The third part of the figure shows the structure of a message "request" sent by the agent "CA-SAMU" to the "CA-SAMU" agent.

6 SOME RELATED WORK

Several models have been proposed to establish the enterprise collaboration. The proposed works have the information system at the center of their concerns and qualify it as an indispensable support for the implementation of the company strategy. The authors of some research work (Indulska et al, 2006) (Henderson et al, 2007) (Rolland et al, 2009) were attached to the analysis and design domain of process-oriented information systems. Several methods and tools of information systems engineering have emerged such as component-based approaches, MDA (Model Driven Engineering). These methods have been proposed to meet the evolution of the current information system context: evolution in terms of the information system architecture (heterogeneous and distributed), evolution in its use (open and accessible for different users). Since a few years the concept of Service Oriented architecture (SOA: Service Oriented Architecture) has been rapidly spread and widely accepted as a supporting architecture of the company information system. We studied the overall work according to the approaches and technologies used the nature of collaborative processes if it is static or dynamic, and the field possible for the use of the proposed solution.

(Touzi et al, 2008) adopt a non standardization approach, which proposes the design of a mediation system between information systems. The proposed practices are similar to the MDE and allow one to anticipate the tooling of a model transformation workshop. This method was also adopted in (Truptil et al, 2011), to solve a crisis management problem, the approach used in (Hui et al, 2010) is a standardization approach that consists on the proposition of a framework where all the participating company must respect it.

Among the works that have used the agent paradigm for solving the inter-company cooperation problems, the authors, (Brahimi et al, 2009) (Namin et al, 2006) have proposed global architectures, integrating the technology of web services. The difference that (Namin et al, 2006) proposed the integration of web services and software agents within the internal structure of the company and by adopting software agents within the UDDI registry; from the company side. Few studies have treated the notion of dynamic cooperation. For instance, the work of (Boukadi et al, 2010) offers such a mechanism of cooperation. (Mallek et al, 2011) have a different vision. They proposed an approach to detect problems before a real collaboration, then analyzing and finding solutions to each partner. A few studies cited above that have used software agents and they have not addressed the decision-making of these agents. In this work we propose a distributed system, and when it comes to designing this type of system, agent technology is suitable, because MAS not only allow the sharing or distribution of knowledge, but also the achievement of a common goal. Agents in our work learn the decision from the user in certain situations to support him in other situations. The decision problem in the case of the cooperative distributed systems is a complex problem. Then, the training algorithm of the agent must meet several requirements: interactive, explicative, online, not handling only the quantitative but also the qualitative parameters. Also we must divide our decision problem into several problems. In our case, we choose to use decision trees, because they can model simply graphically and quickly a complex measured phenomenon. The aim of the use of training agents with the paradigm of decision trees is to select the best services that meet customer needs.

7 CONCLUSIONS

In this paper, we have proposed an architecture to create a dynamic process of cooperation; our architecture can be used in the case of B2B (business-to-business), B2C (Business-to-Consumer) or B2A (business-to- administration). This solution is based on approach of no-standardization; companies keep their structure and offer services. The mediator is involved in making the transformation between the services to create an intercompany collaborative process. We have integrated decision trees in a module of agents called representation module. The use of decision trees can reduce the response time to choose the best service. In order to better improve our approach, we will use the reinforcement learning (RL) algorithm, which has long been promising methods for enabling an autonomous agent to improve its behavior on sequential decision-making tasks. The next step is to specify a transformation model for integrating the decision trees in the agent modules.

REFERENCES

Boukadi, K., Vincent, L., Ghedira, C., Maamar, Z., 2010. CSMA: Context-Based, Service-Oriented Modeling and Analysis Method for Modern Enterprise Applications. In *International Journal of Organizational & collective Intelligence*, pp 1-28.

Brahimi, M., Boufaida, M., Seinturier, L., 2009 Multi-Agent Architecture for Developing Cooperative E-Business Applications. In *International Journal of Information Systems and Supply Chain Management (IJISSCM)*.

Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C., 1984. Classification and Regression Trees, Wadsworth. Inc., Belmont, California.

Fodor, J., 1983. *The Modularity of the Mind*, Cambridge, Mass, MIT Press.

Grefen, P., Kouvas, N., Weichhart, G., Eshuis, R., 2009. Dynamic business network process management in instant virtual enterprises, Computers inIndustry, pp. 86–103.

Henderson, S., Gonzalez, C-P., Ralyté, J., 2007 Situational method engineering: chunks or fragments. In the CAiSE ForumTrondheim, pp 89-92.

Hui, L., Anne, F. Cutting, D., Bourey, J.P., 2010. Use of Ontology for Solving Interoperability Problems between Enterprises. In the *11th IFIP Working Conference on virtual enterprises*.

Indulska, M., Green, P., Recker, J., Rosemann, M., 2006. Business Process Modeling: Perceived Benefits. In the *28th International Conference on Conceptual Modeling* Springer, Gramado, Brazil .

Mallek, S., Daclin, N., Chapurlat, N., 2011. An Approach for Interoperability Requirements Specification and Verification. In *Third International IFIP Working Conference*, pp 89-102.

Namin, A-S., Shen, W., Ghenniwa, H., 2006. Implementing enterprise collaboration using web services and software agents. In *International Journal of Nerworking and Virtual Organisations*, pp 185-201.

Quinlan, J. R., 1984. C4.5: Programs for Machine Learning, Morgan Kaufmann.

Rolland, C., 2009. Method engineering: towards methods as services, Software Process: *Improvement and Practice*, 14(3), pp. 143-164.

Touzi, J., Bénaben, F., Pingaud, H ., 2008. Prototype to Support Morphism between BPMN Collaborative Process Model and Collaborative SOA Architecture Model. In the *4th International Conference on Interoperability for Enterprise Software and Applications*, pp 145-157.

Truptil, S., Bénaben, F., Pingaud., P., 2011. On-the-fly adaptation of crisis response information system. In *Conference on Management of Emergent Digital EcoSystems*. pp114- 121.